

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-099433  
 (43)Date of publication of application : 05.04.2002

(51)Int.CI. G06F 9/46  
 G06F 1/32  
 G06F 1/04

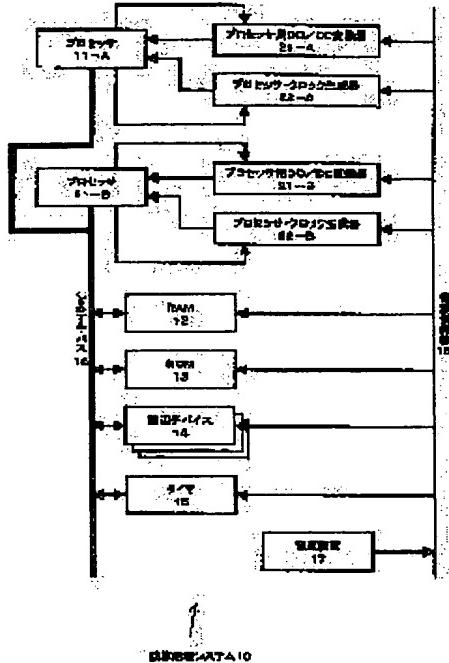
(21)Application number : 2000-287883 (71)Applicant : SONY CORP  
 (22)Date of filing : 22.09.2000 (72)Inventor : TOGAWA ATSUSHI

## (54) SYSTEM OF COMPUTING PROCESSING, CONTROL METHOD SYSTEM FOR TASK CONTROL, METHOD THEREFOR AND RECORD MEDIUM

### (57)Abstract:

**PROBLEM TO BE SOLVED:** To provide a configuration system of multi-processor capable of reducing power consumption of a processor along with responding to real-time demands from applications.

**SOLUTION:** In the configuration system having a plurality of processors enabling to change dynamically an operating frequency and a power source voltage by controlling of an operating system, the power consumption of each processor and a whole of the system is reduced with determining the most suitable power source voltage responding to a time-varying operating frequency in conjunction with adaptable changing of the frequency necessary to perform a started periodical real-time task and a non real-time task without delay.



### LEGAL STATUS

- [Date of request for examination]
- [Date of sending the examiner's decision of rejection]
- [Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]
- [Date of final disposal for application]
- [Patent number]
- [Date of registration]
- [Number of appeal against examiner's decision of rejection]
- [Date of requesting appeal against examiner's decision of rejection]
- [Date of extinction of right]

**Japanese Publication for Unexamined Patent Application**  
**Tokukai 2002-99433 (P2002-99433A)**

**A. Relevance of the above-identified Document**

This document has relevance to claims 1 to 4 and 13 to 16 of the present application.

**B. Translation of the Relevant Passages of the Document**

[0025]

[MEANS TO SOLVE THE PROBLEMS]

...a processing system or a processing control method including a plurality of processors capable of concurrently executing a periodical real-time task that needs to be completed before a next period is started and a non-real-time task that has no limitation as to a completion time, the system or the method characterized by including: operating frequency calculating means or step for calculating, with respect to each processor at each moment during operation, an operating frequency sufficient for meeting a requirement to each periodical real-time task; processor clock generating means or step for supplying, to each processor, an operating frequency clock that is based on a result of calculation in the operating frequency calculating means or step; line voltage calculating means or step for calculating, with respect to each

processor, a line voltage necessary for driving the processor at the frequency calculated in the operating frequency calculating means or step; and processor power supplying means or step for supplying, to each processor, the line voltage that is based on a result of calculation in the line voltage calculating means or step.

[0028]

According to the processing system and the processing control method of the first aspect of the present invention, an adaptive change is made, with respect to each processor, to an operating frequency necessary for the processor so as to process without delay the periodical real-time task and the non-real-time task that are started. Meanwhile, an optimal line voltage is determined with respect to each processor in accordance with the operating frequency, which is constantly changed. With this arrangement, power consumption of each processor is reduced.

[0029]

The processing system and the processing control method of the first aspect of the present invention may further include task choosing/executing means or step for (i) selecting, at each moment during operation of each processor, which given task is to be executed, and (ii) executing the selected task. More specifically, in each processor, a task is fetched from a top of executable periodical real-time tasks, or, if the list is empty, from a top of a list of executable non-real-time tasks. In this way, each processor can execute tasks as required, while meeting requirements to periodical

real-time tasks.

[0030]

Moreover, the system and the method further include transition-to-sleep means or step for causing a processor to be into a sleep mode if at any moment during operation of the processor there is no task to be executed, the sleep mode being a mode in which an operating rate of the processor is lowered. By causing the processor with no unfinished task into the sleep mode, power consumption of the system as a whole is minimized.

(19) 日本国特許庁 (JP)	(12) 公開特許公報 (A)	(11)特許出願公開番号 特開2002-99433 (P2002-99433A)
(43) 公開日 平成14年4月5日(2002.4.5)		
(51) Int.Cl. G 06 F 9/46	範例記号 3 4 0	F 1 G 06 F 9/46 3 4 0 B 5 B 0 1 1 3 4 0 E 5 B 0 7 9 1/04 3 0 1 C 5 B 0 9 8 1/00 3 3 2 Z

## 審査請求 未請求 請求項の数16 OI (全20頁)

(21) 出願番号 特願2000-287883(P2000-287883)

(22) 出願日 平成12年9月22日(2000.9.22)

(71)出願人 000002185  
ソニー株式会社  
東京都品川区北品川6丁目7番35号  
戸川 敦之  
(72)発明者 戸川 敦之  
東京都品川区北品川6丁目7番35号 ソニ  
ー株式会社内  
(74)代理人 100101801  
弁理士 山田 美治 (外2名)  
Fターム(参考) S0011 LL02 LL13  
580793 BA01 S001 B001 D020  
58098 FP03 FF08 GA02 GA04

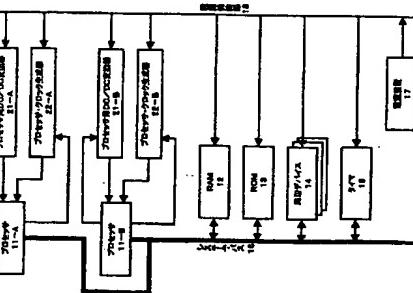
(54) [発明の名前] 電算処理システム及びタスク管理方法、並びに記憶媒体

## (57) [要約]

【課題】 アプリケーションのリアルタイム要求に応えつつプロセッサによる電力消費を削減することができるマルチプロセッサ構成システムを提供する。

【解決手段】 動作周波数と電源電圧をオペレーティング・システムの初期により動的に変化させることができるものである。各プロセッサは複数個えたマルチプロセッサ構成システムにおいて、各プロセッサ毎に、起動された各タスクを逐次なく処理するために必要な動作周波数を適切に変化させるとともに、時々刻々と切り替わる動作周波数に応じて最適な電源電圧を設定していくことで、各プロセッサ及びシステム全体の消費電力を低減する。

【課題】 電算処理システム及びタスク管理方法、並びに記憶媒体



【課題】 1) 次の周期の開始までに実行を行ったときに、各周開リソース毎に、稼動中の各時点において、各周開リソース毎に、動作周波数算出ステップに於ける動作周波数を算出する動作周波数算出ステップと、各周開リソース毎に、前記動作周波数算出ステップにより算出された動作周波数を該プロセッサを駆動させるため結果に基づく動作周波数クロックを該プロセッサによる算出結果に基づく動作周波数クロックを該プロセッサに供給するプロセッサ・クロック生成ステップと、各プロセッサ毎に、前記動作周波数算出ステップにより算出された動作周波数を該プロセッサを駆動させるため充分な動作周波数を算出する動作周波数算出手段と、各周開リソース毎に、前記動作周波数算出手段による算出結果に基づく動作周波数クロックを該プロセッサに供給するプロセッサ・クロック生成手段と、各周開リソース毎に、前記動作周波数算出手段により算出された動作周波数で該プロセッサを駆動させるために充分な電源電圧を算出する電源電圧算出手段と、各周開リソース毎に、前記動作周波数算出手段による算出結果に基づく電源電圧を該プロセッサに供給することを特徴とする算出電源供給ステップと、各プロセッサ毎に、前記電源電圧を該プロセッサによる算出結果に基づく電源電圧を該プロセッサに供給するプロセッサ電源供給ステップと、を具備することを特徴とする演算処理制御方法。

【課題】 2) 各プロセッサの稼動中の各時点において、与えられたタスクのうちそれを実行すべきかを選択して実行するタスク選択・実行ステップをさらに備えることを特徴とする請求項1に記載の演算処理制御方法。

【課題】 3) 各プロセッサの稼動中の各時点において、与えられたタスクのうちそれを実行すべきかを選択して実行するタスク選択・実行手段に備えることを特徴とする請求項1に記載の演算処理システム。

【課題】 4) 各プロセッサの稼動中の各時点において、与えられたタスクのうちそれを実行すべきかを選択して実行するタスク選択・実行手段をさらに備えることを特徴とする請求項1に記載の演算処理システム。

【課題】 5) 各周開リアルタイム・タスクに課せられた前記タスク情報配賦ステップにより記録された情報を参照して、電源電圧と動作周波数が前記プロセッサの限界を超えて、及び／又は、既登録の周開リアルタイム・タスクに課せられた要求が満たされたときに、新たに周開リアルタイム・タスクの登録を制限するタスク登録制御ステップと、を各プロセッサ毎にさらに備えることを特徴とする請求項6に記載の演算処理制御方法。

【課題】 6) 各プロセッサ毎に、非リアルタイム・タスクが存在しないときに前記動作周波数算出ステップによって算出された動作周波数及び／又は前記電源電圧算出ステップによって算出された動作周波数及び／又は前記電源電圧を下限値によって算出された動作周波数及び／又は前記電源電圧を所定の下限値を回復したときに応答して、前記プロセッサの動作周波数及び／又は電源電圧を所定の下限値に設定する下限値設定手段をさらに備えることを特徴とする請求項6に記載の演算処理システム。

【課題】 7) 次の周期の開始までに実行を行ったときに必要がある周開リアルタイム・タスクと実行完了時刻の制約がない非リアルタイム・タスクを並行して実行可能な複数個のプロセッサによる演算処理を制御する演算処理システムであって、各周開リソース毎に、各周開リアルタイム・タスクに記せられた次の周期の開始時間までに実行を行ったときに必要がある周開リアルタイム・タスクを並行して実行可能な複数個のプロセッサのタスクを管理するタスク管理システム。





12	され、本発明の第1及び第2の各側面に係る前記処理システム及び前記算算處理制御方法、タスク管理システム及びタスク管理方法と同様の作用効果を得ることができる。
100421	本発明のさらには他の目的、特徴や利点は、後述する本発明の実施例や添付する図面に基づくより詳細な説明によつて明らかになるであろう。
100431	【発明の実施の形態】以下、図面を参照しながら本発明の実施例を詳解する。
100441	システム構成
10	図11には、本発明の実施に供されるマルチプロセッサ構成の演算処理システム1.0のハードウェア構成を模式的に示している。同図に示すように、演算処理システム1.0は、プロセッサ1-1A及び1-1Bと、RAM(Random Access Memory)1-2と、ROM(Read Only Memory)1-3と、周辺デバイス1-4と、タイマ1-5とを含む。
20	同図に示す演算処理システム1.0は、プロセッサ1-1は2個しか装備していないが、3個以上のプロセッサ1-1を周辺デバイス1-4と、タイマ1-5を含む。周辺デバイス1-4は、3個以上のプロセッサを駆動するマルチプロセッサ構成であっても、同様に各種のプログラムコードを実行することを理解されたい。
100451	【0451】プロセッサ1-1A及び1-1Bは、演算処理システム1.0の主コントローラであり、それとOSの制御下で、それをオペレーティング・システム(OS)の制御下で、各種のプログラムコードを実行するようになってい
100461	【0461】オペレーティング・システムがプログラム実行を管理・制御する単位は、一般に「タスク」と呼ばれる。本実施例に係るプロセッサ1-1A及び1-1Bは、異なる周期で動作する複数のタスクを同時に実行する。
30	タスクは、異なる周期で動作する複数のタスクを同時に実行するものとする。タスクは、異なる周期で動作する複数のタスクを構成しているものとする。タスクは、次の周期の開始までに実行を完了させる必要がある。タスクは、「周期リアルタイム・タスク」と、このような実行完
100471	了時間がない「非リアルタイム・タスク」として、各記述タスクは、タスク実行前に記録する情報を参照して、起動処理と動作用周波数が該プロセッサの限界を超えて、及びノンバス、既登録の周期リアルタイム・タスクと、を具備することを特徴とする記録媒体である。
100481	【0481】本発明の第3及び第4の各側面に係る記録媒体は、例えば、様々なプログラムコードを実行可能な通用性のコンピュータ・システムに対して、コンピュータ・ソフトウェアをコンピュータ上で動作可能な形式で物理的に可能である。
100491	【0491】このようないわゆる記録媒体は、コンピュータ・システム上で所定のコンピュータ・ソフトウェアの機能を実現する機能、コンピュータ・ソフトウェアと記録媒体との接続上の協調的関係を定義したものである。換言すれば、本発明の第3及び第6の各側面に係る記録媒体を介して所定のコンピュータ・ソフトウェアをコンピュータ・システムにインストールすることによって、コンピュータ・システム上では協調的作用が發揮される。



(11)

19  
リアルタイム・タスク<sub>1</sub>を実行することができない状態

なので、登録失敗とする。

【0107】ついで、現在オペレーティング・システムに登録され且つ起動要求が発行された周期リアルタイム・タスクの処理負荷を周期で割った値に合計値<sub>1</sub>と、アリケーション・プログラムによって設定された動作周波数の下限値<sub>1</sub>min,p(うち、大きい値を変数<sub>1</sub>)に設定する。(ステップS 3.3)。【0108】図6には、周期リアルタイム・タスクを登録解除するインターフェースにおいて実現される処理手順を示す。アリケーション・タスク<sub>1</sub>を追加して実行することができる状態なので、r<sub>p</sub>の値を更新して(ステップS 1.3)、登録成功とする。【0109】図6には、周期リアルタイム・タスク<sub>1</sub>を登録解除するインターフェースにおいて実現される処理手順を「に設定とともに、プロセッサ<sub>1</sub>が受け取ったものが何かを、スケジューラ又は待ち行列に並んで登録する(ステップS 2.1)。【0110】タスク<sub>1</sub>がプロセッサ<sub>1</sub>自身のものである場合は、登録削除すべき周期リアルタイム・タスクにタスク番号<sub>1</sub>を付与する(ステップS 2.2)。【0111】ついで、現在オペレーティング・システムに登録されている周期リアルタイム・タスクの処理負荷を周期で割った値の合計値<sub>1</sub>r<sub>p</sub>、並びに、現在オペレーティング・システムに登録され且つ起動要求が発行された周期リアルタイム・タスクの処理負荷を周期で割った値の合計値<sub>1</sub>r<sub>p</sub>から、登録削除を行う周期リアルタイムタスク<sub>1</sub>の分c<sub>1</sub>/r<sub>p</sub>を取り除く(ステップS 2.3, S 2.4)。【0112】また、登録削除を行う周期リアルタイム・タスク<sub>1</sub>が既にスケジューラ内の実行可能リアルタイム・タスク<sub>1</sub>に登録されている場合には、タスク<sub>1</sub>をリストR<sub>r,1</sub>中から削除する(ステップS 2.5)。

【0113】そして、別途定義された再スケジューリング処理を行うことで(ステップS 2.6)、本処理ルーチン全体を終了する。再スケジューリング処理については後に詳解する。

【0104】他方、プロセッサ<sub>1</sub>自身のものではない場合は、他のプロセッサへ登録削除要求を転送して(ステップS 2.7)、該要求の処理完了を待って(ステップS 2.8)、本処理ルーチン全体を終了する。【0105】図7には、周期リアルタイム・タスクを起動開始するインターフェースにおいて実現される処理手順を示す。このフローチャートの形式で示している。以下、このフローチャートに従って、①番目のプロセッサ<sub>1</sub>において周期リアルタイム・タスクの起動開始処理について説明する。【0106】まず、起動開始すべき周期リアルタイム・タスクにタスク番号<sub>1</sub>を付与し(ステップS 3.1)、現在オペレーティング・システムに登録され且つ起動要求が発行された周期リアルタイム・タスクの処理負荷を周期で割った値に合計値<sub>1</sub>r<sub>p</sub>を加

(12)

21

【0117】そして、別途定義された再スケジューリング処理を行うことで(ステップS 4.5)、本処理ルーチン全体を終了する。

【0118】図9には、プロセッサ<sub>1</sub>において非リアルタイム・タスクを登録するインターフェースにおいて実現される処理手順をフローチャートの形式で示していく。

以下、このフローチャートに従って非リアルタイム・タスクの登録処理について説明する。

【0119】まず、新規登録する非リアルタイム・タスクにタスク番号<sub>1</sub>を付与する(ステップS 5.1)。そして、このタスク<sub>1</sub>を、プロセッサ<sub>1</sub>におけるスケジューラ内の実行可能な非リアルタイム・タスクのリストR<sub>b,p</sub>に追加登録する(ステップS 5.2)。

【0120】そして、別途定義された再スケジューリング処理を行うことで(ステップS 5.3)、本処理ルーチン全体を終了する。再スケジューリング処理については後に詳解する。

【0121】図10には、プロセッサ<sub>1</sub>において非リアルタイム・タスクを登録削除するインターフェースにおいて実現される処理手順をフローチャートの形式で示している。以下、このフローチャートに従って、非リアルタイム・タスクはタイムで起動されるものとしたが、本発明の要旨は必ずしもこれに限定されない。最小起動間隔が指定の周期を下回らない範囲で、自由な時刻に周期リアルタイム・タスクを起動するようにして本発明を示す。【0122】まず、登録削除する非リアルタイム・タスクにタスク番号<sub>1</sub>を付与する(ステップS 6.1)。【0123】このタスク<sub>1</sub>が、プロセッサ<sub>1</sub>のスケジューラ内の実行可能な非リアルタイム・タスクのリストR<sub>b,p</sub>中に既に登録されている場合には、該リストR<sub>b,p</sub>中から削除する(ステップS 6.2)。

【0124】そして、別途定義された再スケジューリング処理を行うことで(ステップS 6.3)、本処理ルーチン全体を終了する。再スケジューリング処理については後に詳解する。

【0125】図11には、プロセッサ<sub>1</sub>において周期リアルタイム・タスク<sub>1</sub>が持つ周期<sub>1</sub>を計算した値を次の周期の開始時刻c<sub>1</sub>に代入して(ステップS 3.5)、時刻<sub>1</sub>にタスク<sub>1</sub>が起動するよう【0126】また、登録削除を行う周期リアルタイム・タスク<sub>1</sub>が既にスケジューラ内の実行可能リアルタイム・タスク<sub>1</sub>に登録されている場合には、タスク<sub>1</sub>をリストR<sub>r,1</sub>中から削除する(ステップS 3.6)。【0127】ついで、スケジューラ内の実行可能なリアルタイム・タスク<sub>1</sub>にタスク<sub>1</sub>を追加登録する(ステップS 3.7)。

【0128】そして、別途定義された再スケジューリング処理を行うことで(ステップS 3.8)、本処理ルーチン全体を終了する。再スケジューリング処理については後に詳解する。

【0129】図12には、プロセッサ<sub>1</sub>において周期リアルタイム・タスク<sub>1</sub>が持つ周期<sub>1</sub>を計算した値を次の周期の開始時刻に達したときの処理ルーチン内で行われる再スケジューリング処理の手順をフローチャートの形式で示していく。以下、このフローチャートに従って、再スケジューリング処理について説明する。【0130】まず、起動すべき周期リアルタイム・タスク<sub>1</sub>にタスク番号<sub>1</sub>を付与する(ステップS 4.1)。【0131】ついで、現在時刻にタスク<sub>1</sub>が持つ周期<sub>1</sub>を加算した値を次の周期の開始時刻<sub>1</sub>に代入して(ステップS 4.2)、時刻<sub>1</sub>を設定する(ステップS 4.3)。

【0132】【追加】以上、特定の実施例を参照しながら、本発明の要旨とは直接関連しないので、ここでは省略して説明しない。

【0133】本発明について詳解してきた。しかしながら、本発明の要旨を逸脱しない範囲で当業者が該実施例の修正や代用を成し得ることは自明である。

【0134】本明細書で紹介した実施の形態では、周期リアルタイム・タスクはタイムで起動されるものとしたが、本発明の要旨は必ずしもこれに限定されない。最小起動間隔が指定の周期を下回らない範囲で、自由な時刻に周期リアルタイム・タスクを起動するようにして本発明を示す。

【0135】例えば、ディスプレイ装置を備えた演算処理システムにおいて、垂直駆動期間割り込み信号が発生したときには、タスクを起動する時間が考へられる。また、ネットワークからのバケット割り當時に周期リアルタイム・タスクを起動することも可能である。

【0136】本発明の要旨を判断するためには、冒頭に記載した特許請求の範囲を参考すべきである。

22

ム・タスクのリストR<sub>b,p</sub>が空か否かをチェックする

(ステップS 7.2)。

【0137】リストR<sub>r,p</sub>が空でなければ、該リストR<sub>r,p</sub>中の先頭の非リアルタイム・タスクに周期<sub>1</sub>に開録を移しを付与して(ステップS 7.5)、本処理ルーチン全体を終了する。(ステップS 7.6)スケジューラ内のリストR<sub>r,p</sub>及びリストR<sub>b,p</sub>がいずれも空である場合には、現在プロセッサ<sub>1</sub>が実行すべきタスクがないことになるので、プロセッサ<sub>1</sub>が実行すべきタスクを終了する(ステップS 7.3)。

以下、このフローチャートに従って非リアルタイム・タスクを登録する(ステップS 7.4)。

以下、このフローチャートに従って非リアルタイム・タスクを登録する(ステップS 7.5)。

以下、このフローチャートに従って非リアルタイム・タスクを登録する(ステップS 7.6)。

以下、このフローチャートに従って非リアルタイム・タスクを登録する(ステップS 7.7)。

以下、このフローチャートに従って非リアルタイム・タスクを登録する(ステップS 7.8)。

以下、このフローチャートに従って非リアルタイム・タスクを登録する(ステップS 7.9)。

以下、このフローチャートに従って非リアルタイム・タスクを登録する(ステップS 7.10)。

以下、このフローチャートに従って非リアルタイム・タスクを登録する(ステップS 7.11)。

以下、このフローチャートに従って非リアルタイム・タスクを登録する(ステップS 7.12)。

以下、このフローチャートに従って非リアルタイム・タスクを登録する(ステップS 7.13)。

以下、このフローチャートに従って非リアルタイム・タスクを登録する(ステップS 7.14)。

以下、このフローチャートに従って非リアルタイム・タスクを登録する(ステップS 7.15)。

以下、このフローチャートに従って非リアルタイム・タスクを登録する(ステップS 7.16)。

以下、このフローチャートに従って非リアルタイム・タスクを登録する(ステップS 7.17)。

以下、このフローチャートに従って非リアルタイム・タスクを登録する(ステップS 7.18)。

以下、このフローチャートに従って非リアルタイム・タスクを登録する(ステップS 7.19)。

23

ム・タスクのリストR<sub>b,p</sub>が空か否かをチェックする

(ステップS 7.2)。

【0138】リストR<sub>r,p</sub>が空でなければ、該リストR<sub>r,p</sub>中の先頭の非リアルタイム・タスクに周期<sub>1</sub>に開録を移しを付与して(ステップS 7.5)、本処理ルーチン全体を終了する。(ステップS 7.6)スケジューラ内のリストR<sub>r,p</sub>及びリストR<sub>b,p</sub>がいずれも空である場合には、現在プロセッサ<sub>1</sub>が実行すべきタスクがないことになるので、プロセッサ<sub>1</sub>が実行すべきタスクを終了する(ステップS 7.3)。

以下、このフローチャートに従って非リアルタイム・タスクを登録する(ステップS 7.4)。

以下、このフローチャートに従って非リアルタイム・タスクを登録する(ステップS 7.5)。

以下、このフローチャートに従って非リアルタイム・タスクを登録する(ステップS 7.6)。

以下、このフローチャートに従って非リアルタイム・タスクを登録する(ステップS 7.7)。

以下、このフローチャートに従って非リアルタイム・タスクを登録する(ステップS 7.8)。

以下、このフローチャートに従って非リアルタイム・タスクを登録する(ステップS 7.9)。

以下、このフローチャートに従って非リアルタイム・タスクを登録する(ステップS 7.10)。

以下、このフローチャートに従って非リアルタイム・タスクを登録する(ステップS 7.11)。

以下、このフローチャートに従って非リアルタイム・タスクを登録する(ステップS 7.12)。

以下、このフローチャートに従って非リアルタイム・タスクを登録する(ステップS 7.13)。

以下、このフローチャートに従って非リアルタイム・タスクを登録する(ステップS 7.14)。

以下、このフローチャートに従って非リアルタイム・タスクを登録する(ステップS 7.15)。

以下、このフローチャートに従って非リアルタイム・タスクを登録する(ステップS 7.16)。

以下、このフローチャートに従って非リアルタイム・タスクを登録する(ステップS 7.17)。

以下、このフローチャートに従って非リアルタイム・タスクを登録する(ステップS 7.18)。

以下、このフローチャートに従って非リアルタイム・タスクを登録する(ステップS 7.19)。

(13)

23

【013.8】また、本発明によれば、動作周波数と電源電圧を動的に制御することができるプロセッサを複数備えたマルチプロセッサ構成システムにおいて、各プロセッサ毎に、起動された各タスクを通常なく処理するため必要な動作周波数を適応的に変化させるとともに、時々刻々と切り替わる動作周波数に応じて最適な電源電圧を決定していくことで、各プロセッサ並びにシステム全体の消費電力を低減することができる。優れた電力削減技術を提供することができる。

【013.9】また、本発明に係るマルチプロセッサ構成システムを用いた場合、同一のプロセッサにより同じ処理量のタスクを実行する場合に比し、消費電力を削減することができる。

【014.0】また、本発明によれば、プロセッサの動作周波数や電源電圧の変更頻度を低く抑えながら、各のプロセッサによる電力消費を削減することができる。

【014.1】周期リアルタイム・タスクは次の周期の開始以前に処理を完了するタスクである。処理を完了すべき時間はなわちドライインを登録してある。各のシステムを位置づけるとともに考えられるが、本発明では仮えてこのようにして処理完了時間に制限を課すことによつて、タスクの登録や登録削除時に動作周波数を変化させるだけで十分に高い電力削減効果を得ることができる。

【014.2】したがって、プロセッサの電源電圧や動作周波数を変化させるために比較的長い時間が必要するハードウェアに対しても、本発明を適用することができます。

【014.3】例えば、あるタスク1について、デッドライインを次の周期の開始時間よりも前に設定するがシステム構築上どうしても必要になった場合には、タスク起動タイミングの時刻設定を徐く、変数D1に対するすべての参照をデッドライインに対する参照へと置き換えるだけでよい。

【図面の添付な説明】  
【図1】本発明の実施に供されるマルチプロセッサ構成の演算処理システム10のハードウェア構成を模式的に示した図である。

【図2】複数の周期リアルタイム・タスクが異なる周期で同時に実行されている様子を模式的に示した図である。

(14)

24

【図3】本実施例に係るオペレーティング・システムの機能構成を模式的に示したブロック図である。

【図4】周期リアルタイム・タスクを登録するインターフェースにおいて実現される処理手順を示したフローチャートである。

【図5】プロセッサににおける周期リアルタイム・タスクの登録判定処理の手順を示したフローチャートである。

【図6】周期リアルタイム・タスクを登録解除するインターフェースにおいて実現される処理手順を示したフローチャートである。

【図7】周期リアルタイム・タスクを起動開始するインターフェースにおいて実現される処理手順を示したフローチャートである。

【図8】周期リアルタイム・タスクを起動するタイミングの設定時刻に達したときににおける処理手順を示したフローチャートである。

【図9】非リアルタイム・タスクを登録削除するインターフェースにおいて実現される処理手順を示したフローチャートである。

【図10】非リアルタイム・タスクを登録削除するインターフェースにおいて実現される処理手順を示したフローチャートである。

【図11】非リアルタイム・タスクを登録削除するインターフェースにおいて周期リアルタイム・タスクの登録削除時、起動開始時、非リアルタイムタスクの登録時、登録削除時にを行う再スケジューリング処理の手順を示したフローチャートである。

【符号の説明】

1.0…演算処理システム

1.1…プロセッサ

1.2…RAM

1.3…ROM

1.4…周辺デバイス

1.5…タイマ

1.6…システム・バス

1.7…電源装置

1.8…电源供給線

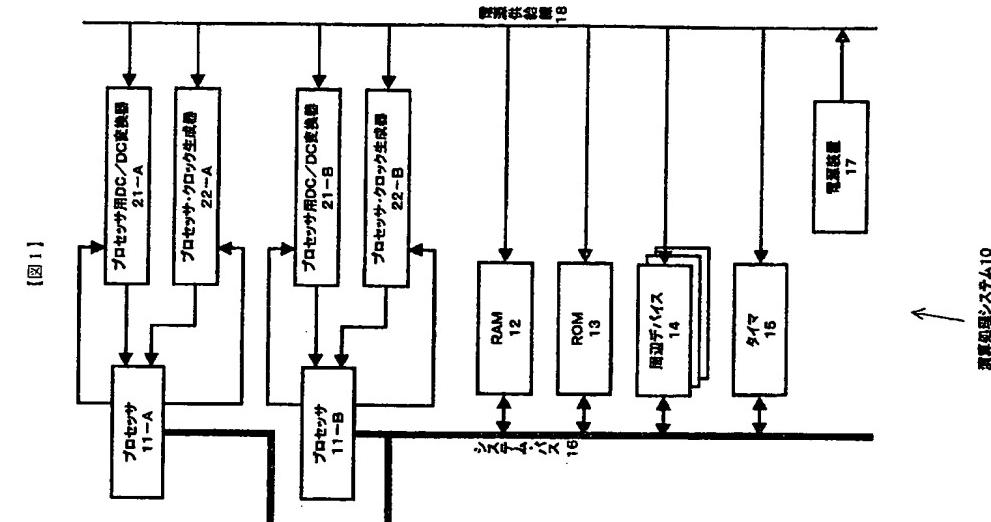
2.1…プロセッサ用DC／DC変換器

2.2…プロセッサ・クロック生成器

(14)

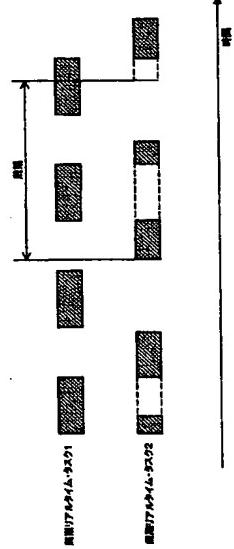
25

【図1】



(15)

[図2]

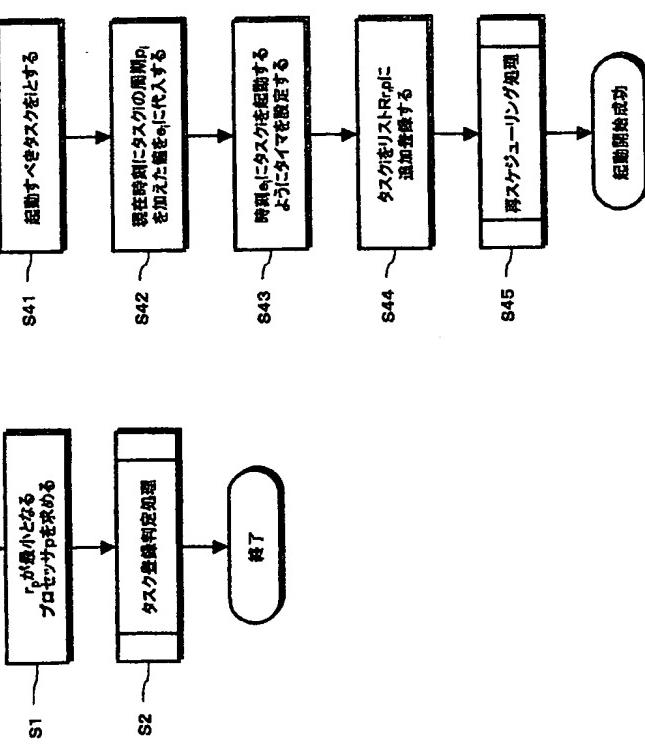


[図4]

周期的アルタイム  
タスクの登録

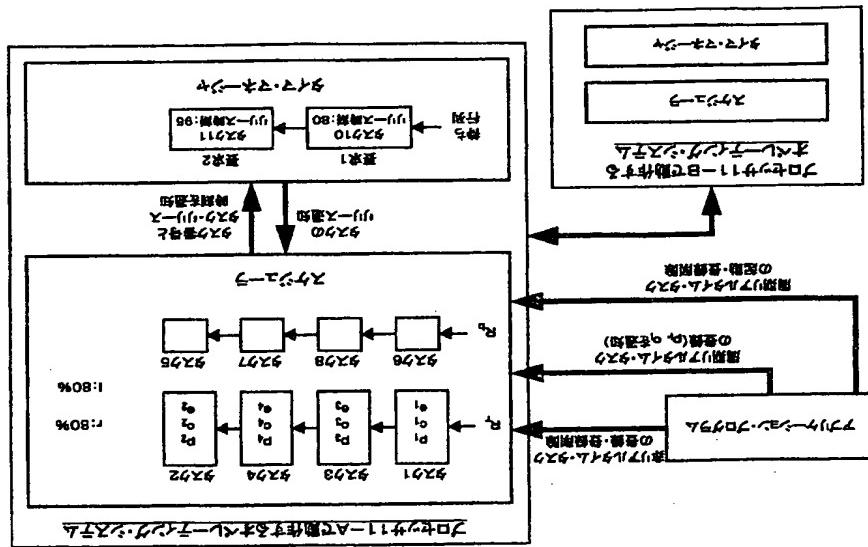
タイム起動例に指定された  
時刻に達したときの処理

[図8]

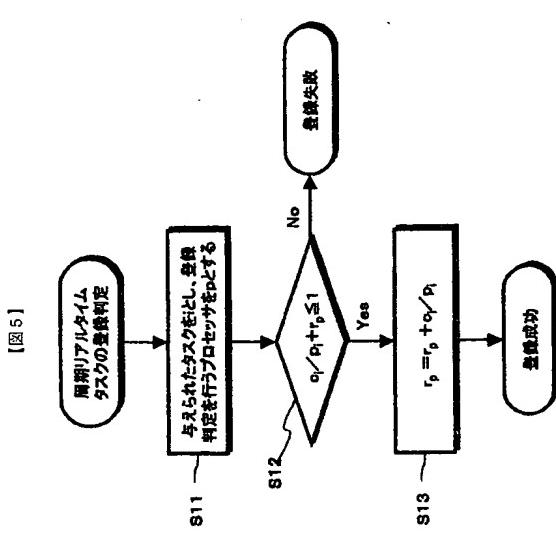


(16)

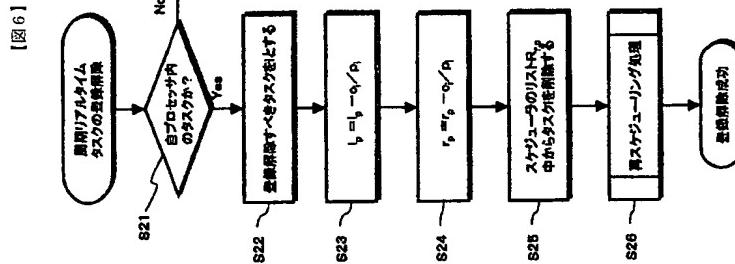
[図3]



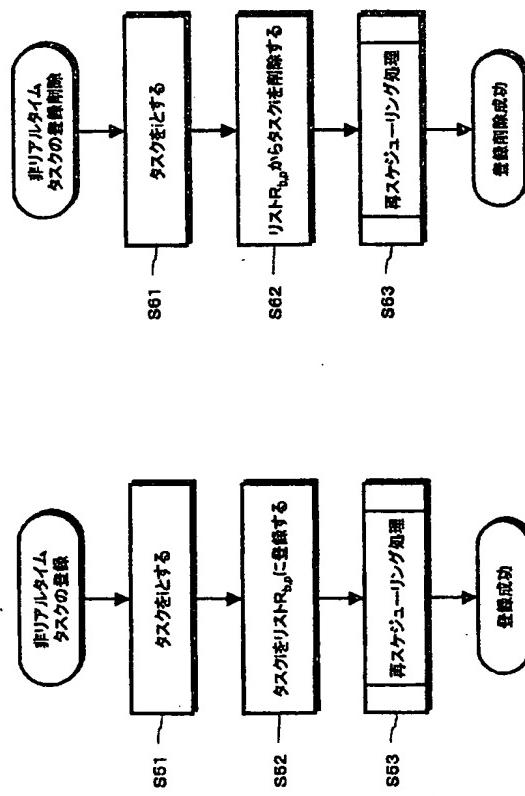
(17)



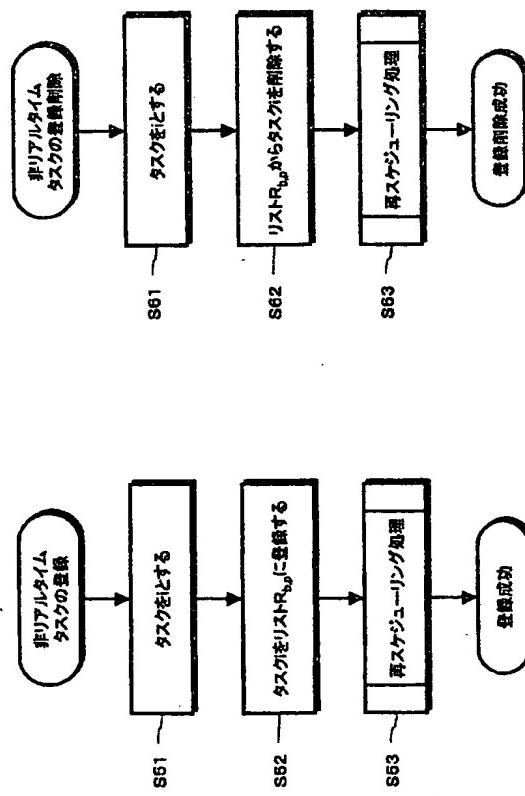
(18)



【図10】



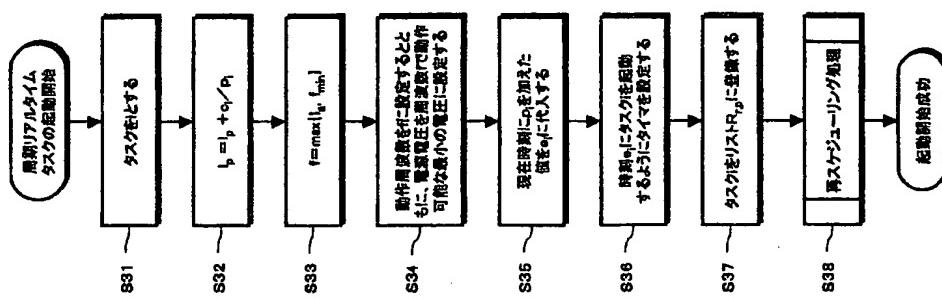
【図10】



(19)

(20)

【図7】



【図11】

